



chipsynth SFC

# ***SUPER INSTRUCTION MANUAL***

v1.023



# INTRODUCTION

It's been a tough year for all of us no doubt about it.

Like many of you I'm sure, these hard times made me reflect on the past and how lucky we were. I've been revisiting many classic albums and games in the last year, and witnessed how much the SNES games had a unique charm.

Its soundtracks are of course what interested me the most. 8 channels of compressed and cleverly looped snippets of sound combined with a truly unique echo/delay was all it had, and yet its music still evoke the same emotions as if it were played by a full orchestra.

While I think we outdid ourselves trying to preserve the unique charm of the SNES's audio subsystem by careful reverse engineering, an instrument is nothing without a usable work-flow. How do we expose the limitations of the SNES audio system in a way that is not too obtuse for the contemporary musician?

That's why the involvement of Mathew Valente [TSSF] was crucial for our project to succeed. Mathew has been doing SNES video game music remakes for over two decades using every SNES music tool under the sun (even some entirely written in Japanese). He knows this stuff inside and out. We were truly fortunate to be able to scratch his brains.

Thank you so much Mathew.

David Viens  
September 14th 2020



## **AUTHORIZING CHIPSYNTH SFC**

You will need to authorize chipsynth SFC to make it fully functional – otherwise it will run in DEMO mode. When you order a license from our web store, you will receive a personal Activation Key card named xxxx\_key\_1101.png as an email attachment.

The Activation Key card is a picture that looks like a typical credit card. This image contains your registration and details encoded within the Key card image. It will look like this:



You should save the xxxx\_key\_1101.png image file to your hard drive (keep it in a safe place). For convenience, we recommend that you initially save the .png file to your desktop.

- 1)** Locate the “license card” image where you saved it on your hard drive. (**xxxx\_key\_1101.png**)
- 2)** Open the chipsynth SFC software application, or launch your favorite host and make sure you see the chipsynth SFC interface.
- 3)** Simply click and hold on the file, and drag-and-drop the “license image” or file icon directly onto the application's UI itself, then release the mouse button.

If you don't get any message (or are not able to drop the key image in that host), try to import the png file from the snapshot load menu or the slot load **import**. (use \*.\* as the file type filter in the dialog)

If all goes well, you should be presented with a message saying “Plogue Art et Technologie, Inc chipsynth SFC is now activated for {your name here}”.

If none of those methods work, please contact [chipsynth.support@plogue.com](mailto:chipsynth.support@plogue.com) and attach your key.

## How to use chipsynth SFC

### *Extremely Important!!*

The xxxx\_**key\_1101.png** file contains your **sensitive personal information**, both encrypted inside and visible in the image itself, including your **full name and address** taken from the online shop. Carefully protect this file. DO NOT GIVE THIS FILE TO ANYONE OR DISTRIBUTE IT IN ANY WAY OR YOUR PERSONAL INFORMATION WILL BE COMPROMISED. IF THE FILE BECOMES PUBLIC THE CARD NUMBER WILL BE BLACKLISTED AND THE CARD REVOKED. WE ARE NOT RESPONSIBLE IF YOU GIVE YOUR PERSONAL DETAILS TO A THIRD PARTY. IF THE CARD IS STOLEN, CONTACT US IMMEDIATELY. Without a valid card you will also not be able to obtain critical updates to the program.

### *Important Note:*

If you have special circumstances or require site licensing, please contact us.

### *Updating to the Latest Version*

Be sure to check the Plogue Web sites for any possible updates that have occurred since the time your version of the software was released. Software is frequently updated and a more recent version may be available.

## Getting Help

The first place to look for a solution to any problem you may be experiencing is in this manual. Please read the manual before contacting support.

Plogue chipsynth SFC is dynamic — evolving and growing. Please check the support area of our website at <https://www.plogue.com> for the latest up-to-date information about products, troubleshooting, FAQs, helpful hints and tutorials. Another good resource is the support forums.

Whenever you encounter problems, you should also check to make sure you have installed the latest updates. The version number of your software is displayed in the **About** dialog. Updates are released regularly to fix known problems and to improve the software.

If you can't find a solution to your problem, please email us at [chipsynth.support@plogue.com](mailto:chipsynth.support@plogue.com). The best way to let us get you the help you need is by giving us plenty of detailed information about the problem you're having. We do ask you to read this guide thoroughly and exhaust the other avenues of support before contacting us.

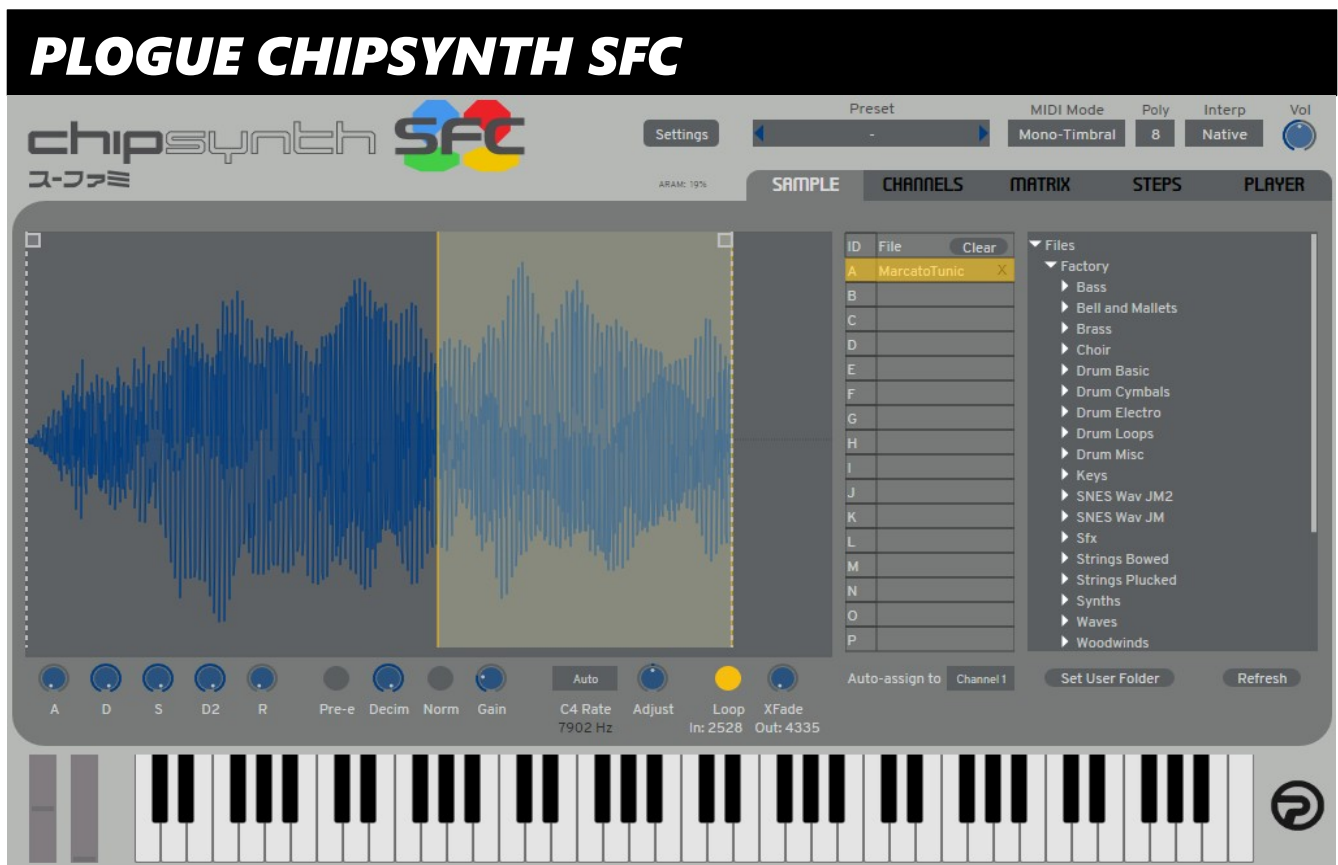
The Plogue forum can be accessed at: <https://www.plogue.com/plgfrms/>  
You don't have to register to browse the forum, but you will have to sign up if you want to post.

*End User License Agreement:* See Licence.rtf, which contains the full license agreement.

Plogue Art et Technologie, Inc.  
[www.plogue.com](http://www.plogue.com)  
[www.facebook.com/wwwPLOGUEcom](https://www.facebook.com/wwwPLOGUEcom)  
[twitter.com/wwwPLOGUEcom](https://twitter.com/wwwPLOGUEcom)



## How to use chipsynth SFC



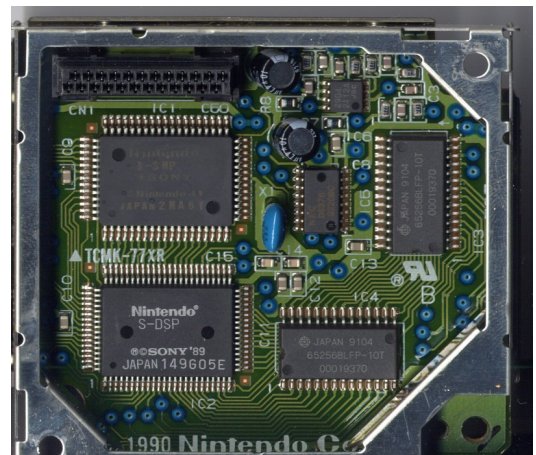
Plogue chipsynth SFC is a multi-layer sampler that faithfully reproduces the sound of the Super Nintendo Entertainment System™ (also known as Super Famicom™), which lets you easily bring back this classic sound in a modern DAW, or stack tons of layers to create a big, lush modern sound. The editing capacities are very extensive to let you use any sample you like as a SFC instrument.

## THE SPC700 SYSTEM

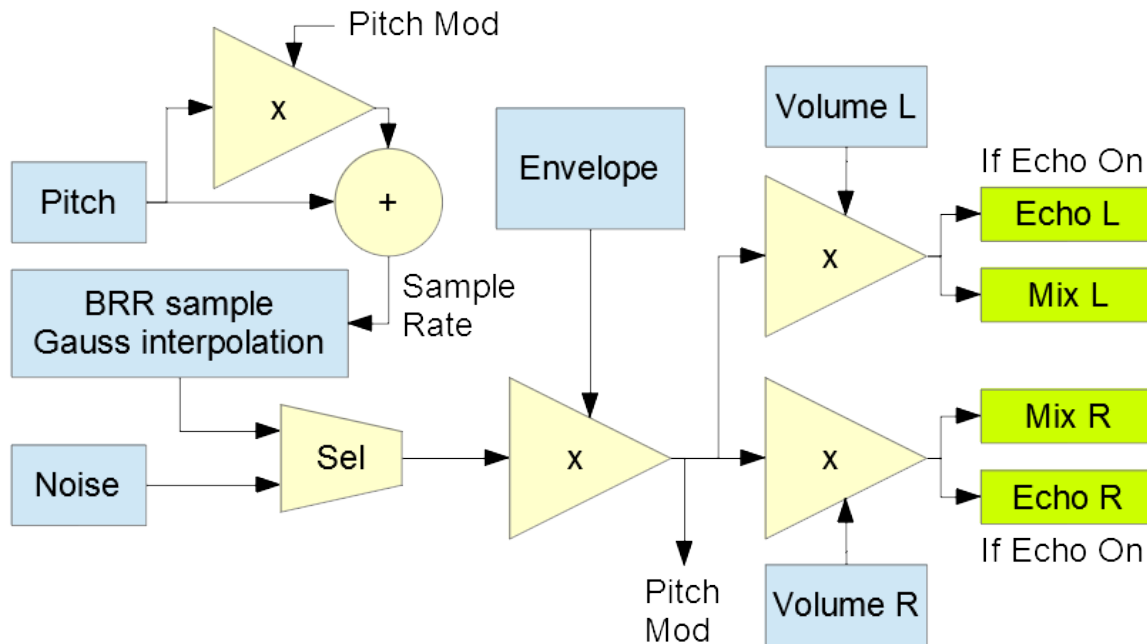
The SNES sound module contains a full set of components:

- The S-SMP (aka SPC700, ≈ 6502+timers and ports)
- The S-DSP chip (responsible for all synthesis)
- 64kb of static RAM (also known as “ARAM”)
- A NEC 16-bit stereo PCM DAC
- A dual op-amp analog filtering section
- Connectors going back to main board

We’ve extensively reverse-engineered this system, to the point that it will exactly match digital recordings of the original perfectly down to the single bit!



## HOW IT WORKS



Each voice plays a compressed sample:

- Note pitch sets the sample rate (with optional pitch modulation).
- BRR-compressed sample data is loaded & interpolated using Gauss interpolation.
- When in noise mode, this gets replaced with white noise.
- An ADSR volume envelope is applied.
- The left and right volumes are applied.
- The sound is mixed into the main mix bus, and optionally into the echo bus.

## GLOBAL SETTINGS



**Presets:** Loads or saves global chipsynth SFC presets.

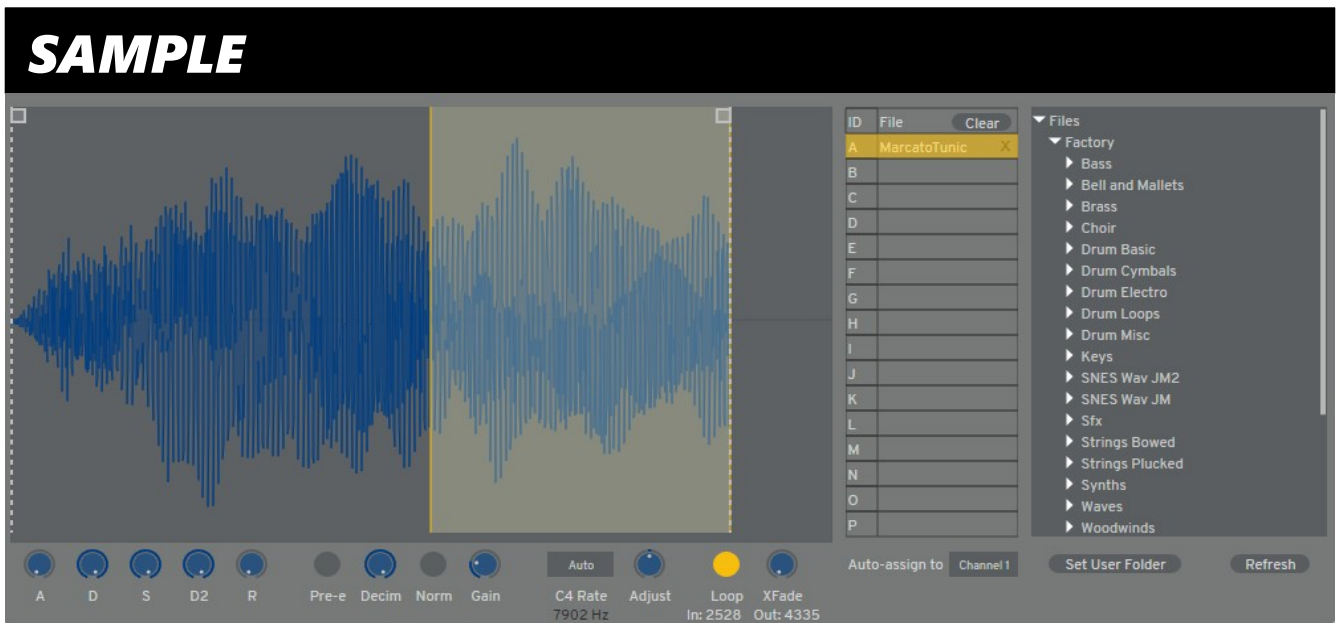
**MIDI Mode:** Selects between mono-timbral (default), poly-timbral and MPE modes.

**Polyphony:** Sets the number of simultaneous notes (1..8). Set this to 1 for monophonic mode.

**Interpolation:** Selects the type of sample interpolation between either the native Gauss interpolation, or the alternative nearest-neighbor, linear or cubic interpolation (which are generally brighter).

**Volume:** Sets the overall volume of chipsynth SFC.

## How to use chipsynth SFC



**Wave display:** Shows the current wave data.

**Active region:** Sets which portion of the wave data is used.

**Loop selection:** Sets the loop start and end region if the loop is active. Chipsynth SFC automatically applies padding and resampling to fit into the SNES's limit of loop points having to be a multiple of 16.

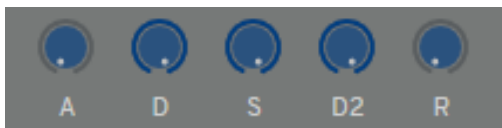
**Sample List:** All the current samples. You can change sample names and delete samples here.

**Auto-assign:** Automatically change samples on channels when browsing samples. Select 'None' to turn this off.

**Sample Bank:** Lets you load samples from the sample bank bundled with chipsynth SFC. This bank contains a broad selection of common instrument samples.

**User Folder:** Sets a folder to use for user samples. You can also load samples using drag-and-drop.

### Envelope Settings



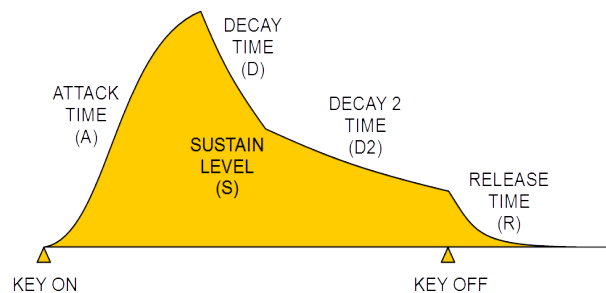
**Attack Time (A):** Initial ramp-up speed.

**Decay Time (D):** Decay speed after the attack.

**Sustain Level (S):** When to switch to decay 2.

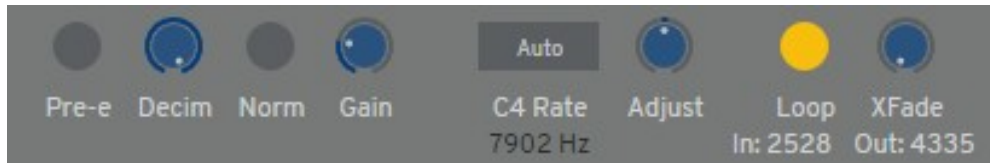
**Decay 2 Time (D2):** Decay speed after sustain level. Set to max (31) for continuous sounds.

**Release (R):** Decay speed when the key is released.



## How to use chipsynth SFC

### Sample Settings



**Pre-Emphasis:** Boosts the sample's high frequencies to balance the loss from Gaussian interpolation.

**Decimate:** Reduce the sample rate of the sample. This increases the range that can be played without octave fallback (the S-DSP cannot play anything at a rate higher than 128khz!).

**Normalize:** Increase sample volume so that the highest peak reaches 100% of the range.

**Gain:** Increases the sample's volume gain. When combined with normalization, gains higher than 100% will create distortion. This effect tends to be somewhat grungy on melodic and bass samples, but works very well on drums.

**C4 Sample Rate Source:** Pick the source for middle C sample rate:

- Auto: Use pitch tracking algorithm to tune the sample to middle C. (default)
- File: Use the rate from the original wave file.
- Custom: Lets you edit the sample rate.

**C4 Sample Rate (Hz):** Sample rate used for middle C. When the source is set to 'custom', this becomes editable.

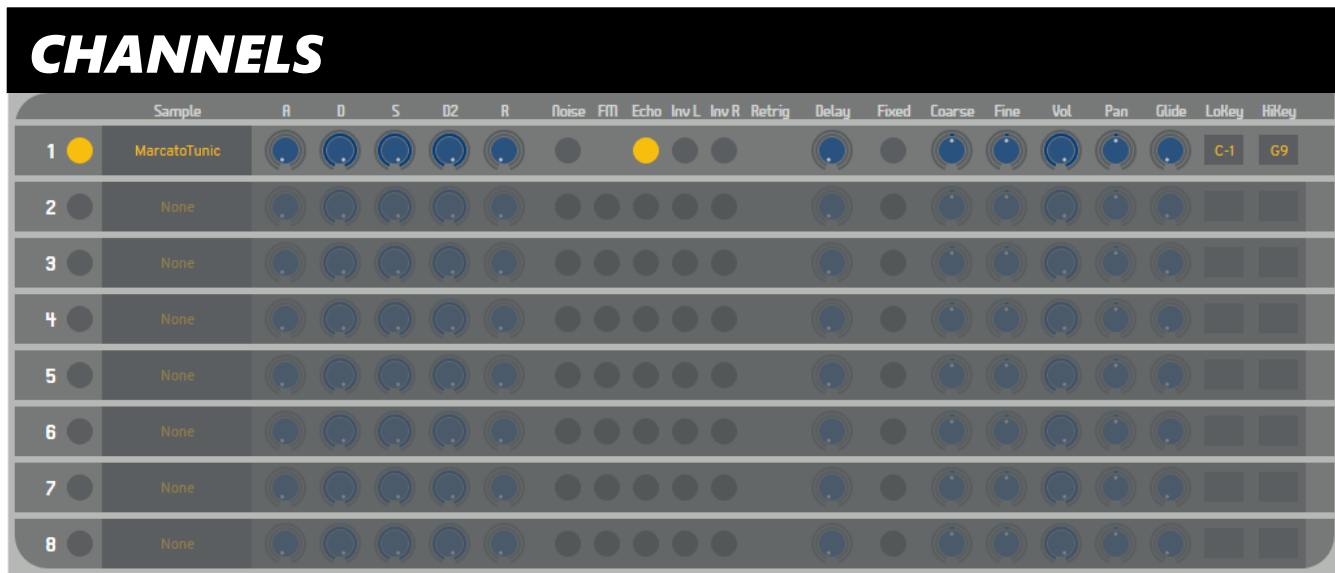
**Adjust Tuning:** Fine tune the sample pitch (in cents).

**Loop:** Activate the sample loop.

**Loop in, Loop out:** Start and end of the loop. You can edit this either graphically by moving the loop selection on the waveform, or numerically by clicking on the loop in/out text displays.

**XFade:** Add a crossfade to the looped audio. Data from before the loop-start will be added to the loop.





This lets you use each of the 8 channels as a synthesis layer, or as a timbre in a polytimbral setup.

**On/off:** Turns the channel on/off.

**Sample Selector:** Selects a sample from slot A to slot P (or “None”, which turns the slot off).

**Envelope Settings (A, D, S, D2, R):** Channel envelope settings. These settings will automatically be replaced when you edit sample envelope settings, load new samples or change the selected sample.

**Noise:** Replace the sample data with noise generator output. Note: you still have to load a sample for the channel to work. Ideally you shouldn’t pick a non-looped sample, as the note will still cut out when the sample ends.

**FM:** Modulate the frequency of this channel with the output of the previous channel. The modulation level is affected by the envelope but not the volume level.

**Echo:** Mix this channel’s output into the echo input bus (in addition to the main mix bus).

**Invert L/R:** Invert the channel’s phase on the left/right side. (can create heavy phase cancellation!)

**Retrig (monophonic mode only):** When doing legatos, fully retrigger the sample and envelope.

**Delay:** Delays the key press events on this voice.

**Fixed freq:** Removes the effect of the keyboard on pitch. MIDI note 36 (C2) is played instead.

**Coarse tune:** Transposes the pitch (in semitones).

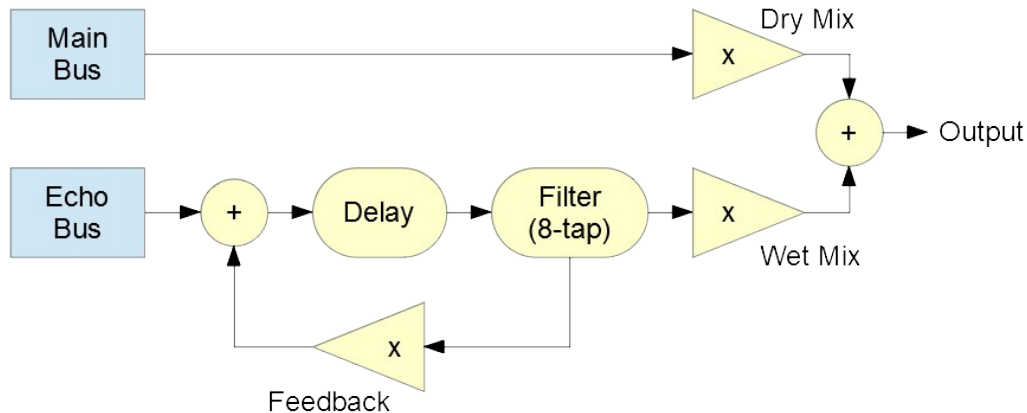
**Fine tune:** Detunes the pitch (in cents).

**Volume:** Sets the overall amplitude of this layer.

**Pan:** Pans the sound in stereo.

**Glide:** Sets the pitch glide time. Setting this to 0 will turn off gliding.

**Lo key, hi key:** Limits the range of keys on which the voice plays.



The echo is a simple delay effect. It's an integral part of the original S-DSP sound module, and is faithfully emulated by chipsynth SFC.

**Preset:** Loads and saves delay presets.

**Time:** Sets the time between repeats in increments of 16ms, up to 240ms.

**Feedback:** Re-injects the signal back into the delay.

**Dry:** Controls the output level of all the non-delayed sound.

**Dry Pan:** Pans the non-delayed sound left/right.

**Wet:** Controls the output level of the delay effect.

**Wet Pan:** Pans the delayed sound left/right.

#### Filter Settings

**0khz, 4khz, 8khz, 12khz, 16khz levels:** Filter gain at 0khz, 4khz, 8khz, 12khz and 16khz respectively.

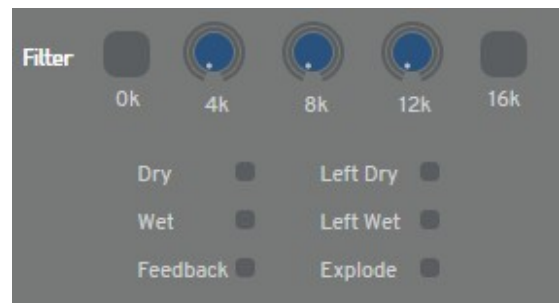
#### Phase Settings

**0khz, 4khz, 8khz, 12khz, 16khz phase:** Phase rotation at 0hz, 4khz, 8khz, 12khz and 16khz. Rotation at 0khz & 16khz can only be 0° or 180°. Affects gain.

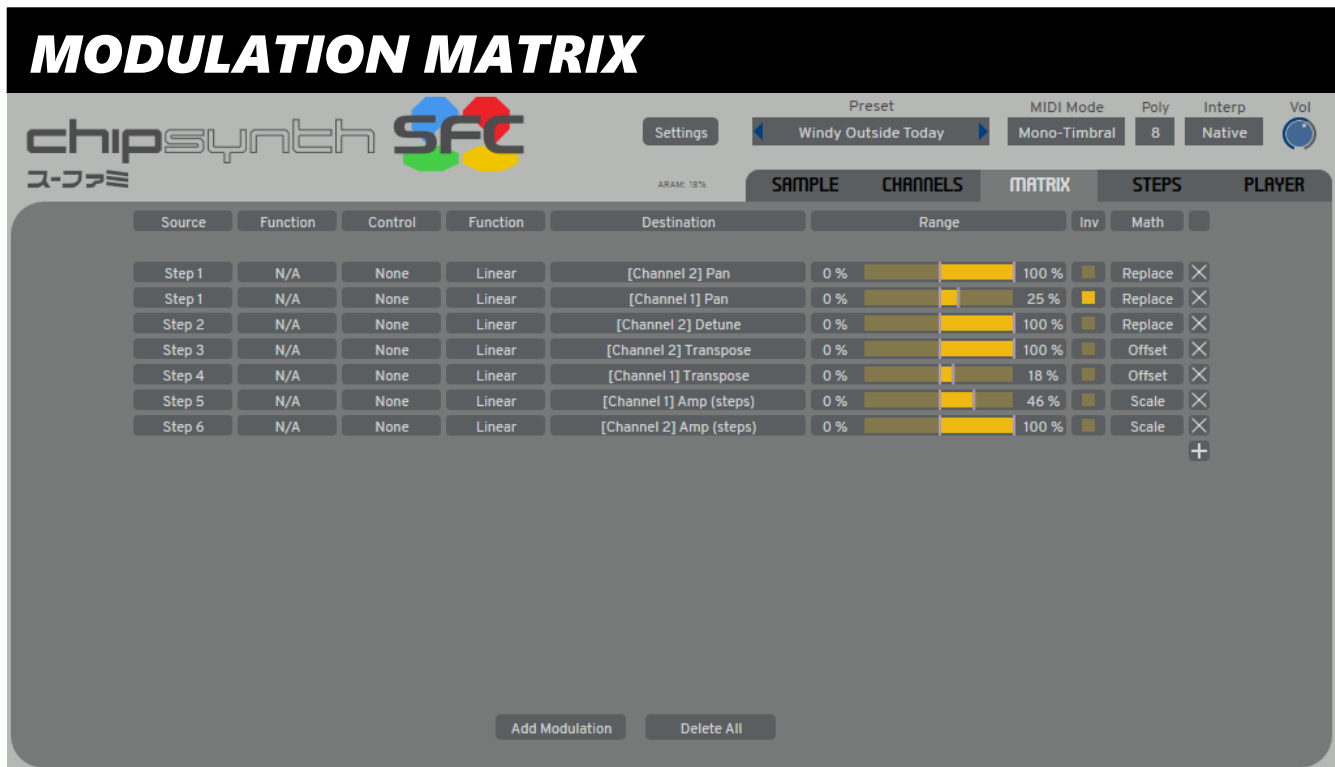
**Dry, Wet, Feedback inversion:** Inverts the phase of the dry, wet and feedback loop respectively.

**Dry, Wet left channel inversion:** Inverts the phase of the left side in the dry and wet out. Causes heavy nulling when downmixing to mono.

**Explode:** Forces the feedback gain to higher than unity, extremely quickly causing an extremely loud numerical explosion of the delay effect. This is not for the faint of heart and will blow up in your face!



## How to use chipsynth SFC



Plogue chipsynth SFC has a Modulation Matrix. This is a powerful tool to add step sequencers, and to let MIDI CCs modulate parameters. You can also assign modulations by right clicking parameter knobs and sliders and selecting the source from the list.

**Source:** Sets which step sequencer or MIDI value modulates the parameter.

**Source Function (MIDI modulation only):** Applies a mathematical function on the source.

**Control:** Adds a second modulation source to scale the amount of modulation (often the Mod wheel).

**Control Function:** Applies a mathematical function on the control.

**Destination:** Selects which parameter gets modulated. Not all parameters are available for every modulator.

**Range:** Scale the modulation into a range. You can only set the upper range on non-MIDI modulators.

**Invert:** Inverts the effect of the modulator.

**Math:** Chooses the operation used to combine value sources. Note: MULTIPLE MODULATION TYPES CANNOT BE USED ON A SINGLE DESTINATION. They must all be 'Offset' or all be 'Scale'.

Math	Effect
Offset	All sources are added together.
Replace	The "base" value is ignored and the modulation takes over. Destinations modulated using this will change according to the LAST updated source.
Scale	All sources are multiplied together.

Special note: you cannot modulate the same destination multiple times with the same source.

## STEP SEQUENCERS

The step sequencers let you add complex custom modulation envelopes to any parameter you want, including a wide variety of channel settings.

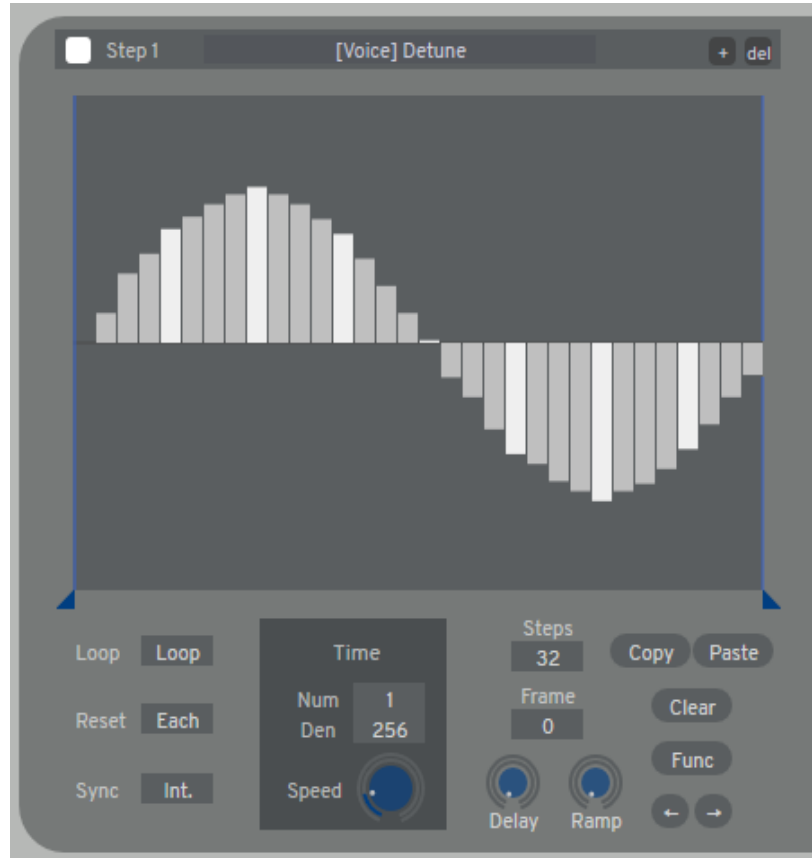
This makes it possible to create evolving/rhythmic sounds, pitch envelopes and so forth. You can even do 'audio rate' modulation with key-tracked frequencies.

By changing the loop mode, both envelopes and LFOs can be created. The envelope is tempo-synced by default, but this can also be changed.

### Sequences

**Display:** Draw the shape of your modulator here!

**Loop Points:** Sets the start and end points of the loop. (You must be in 'Gate' or 'Loop' mode.)



**Loop Mode:** Switches the sequence between one shot, loop with intro and note-off, or loop only.

Loop Mode	Typical use	Effect	Note Off
One Shot	Envelope	No loop points. Plays the full sequence, then stops.	Ignored
Note Gate	Sustained envelope + LFO	Starts at beginning and loops between loop start and end. Jumps to points after loop end on note off.	Jump to after loop
Loop Always	LFO	Loops forever from loop start to loop end. All points before loop start and end are ignored.	Ignored

**Reset Position:** Selects the key sync type.

Reset Position	Type	Synchronization
Each Note	Polyphonic	Resets on every note
First Note	Monophonic	Resets on the first note of a group
Last Note		Resets on every note
Bar Pos		Tick advances are synchronized with music when playing or recording.

## How to use chipsynth SFC

**Sync BPM Source:** Selects the clock source used for the step rate.

Sync Source	Step rate
Host BPM	Follows song BPM.
Internal BPM	Always 125 BPM. Setting the time division to 1:96 gives 50 steps per second.
Note Period	Step duration is scaled to note. 1:128 gives 1 step per period.

**Time Division:** Sets the step rate as a musical duration. Ex: Setting this to 1:16 makes each step last a sixteenth note.

**Speed Scale:** Multiplies the step rate.

**Steps:** Sets the number of sequence steps.

**Frame:** Selects which one of the sub-sequences is used. This can be modulated.

**Delay:** Wait some time before starting the modulation.

**Ramp:** Ramp up the modulation effect rather than applying the full level instantly.

**Copy:** Copy modulator shape to clipboard.

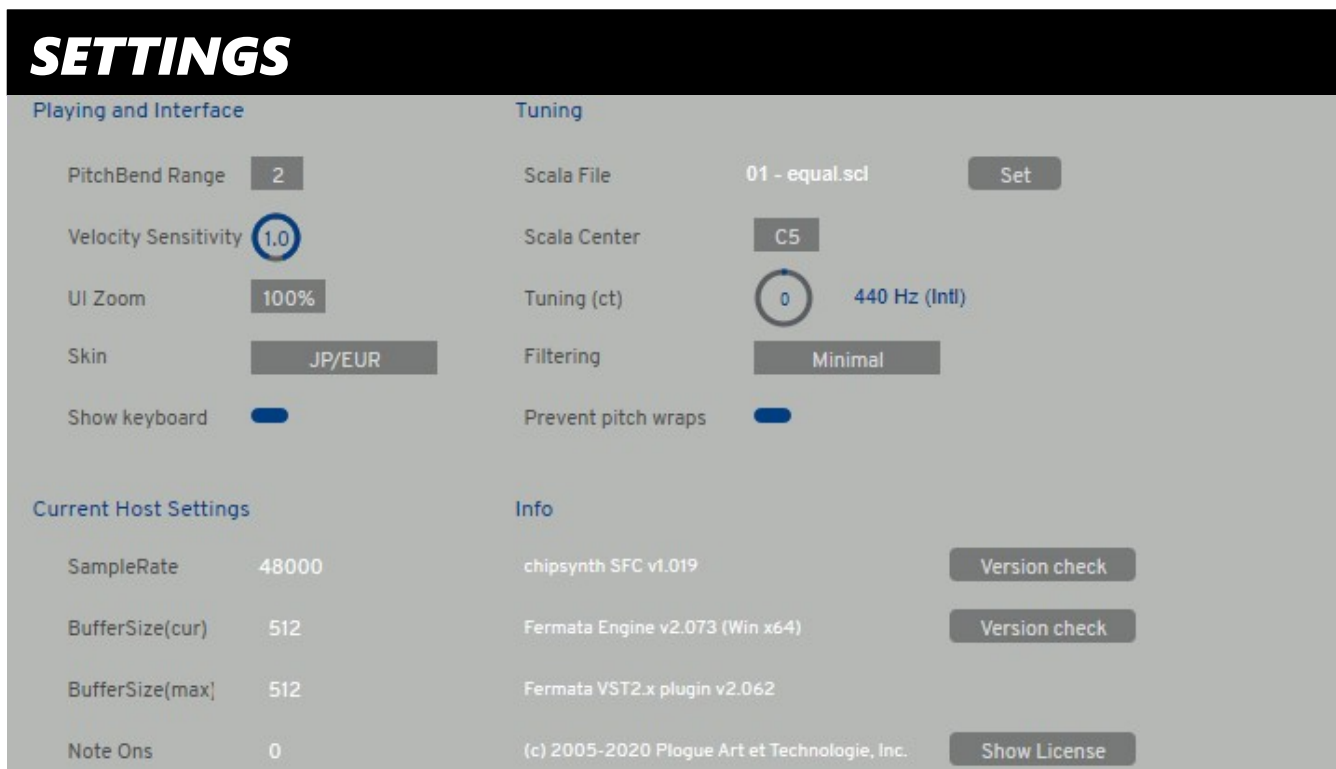
**Paste:** Paste modulator shape from clipboard.

**Clear:** Erase the modulator shape back to blank.

**Func:** Set the modulator shape to one of the preset waveforms.

**Shift left/right:** Rotate the modulator shape by 1 position to the left or right.





## Playing and Interface

**Pitch Bend Range:** Sets the range of pitch bend wheel (in semitones).

**Velocity Tracking (Veltrack):** Controls the global level of the velocity effect on note volumes.

**UI Zoom:** Select the size of the user interface.

**Skin:** Select the color scheme of the interface between a Japan/Europe theme and a US theme.

**Show Keyboard:** Displays the mouse-clickable keyboard under the interface.

## Tuning

**Scala File:** Lets you to load a custom micro-tonal tuning file.

**Scala Center:** Selects the root note from which the Scala tuning is applied.

**Tuning (cents):** Detunes the whole of chipsynth SFC to other frequencies than A 440Hz.

**Filtering:** Selects between a light output filter, or a heavier one modeled after the real analog output.

**Prevent Pitch Wraps:** Stop notes with rates >128kHz (can't play on SPC) instead of octave-wrapping.

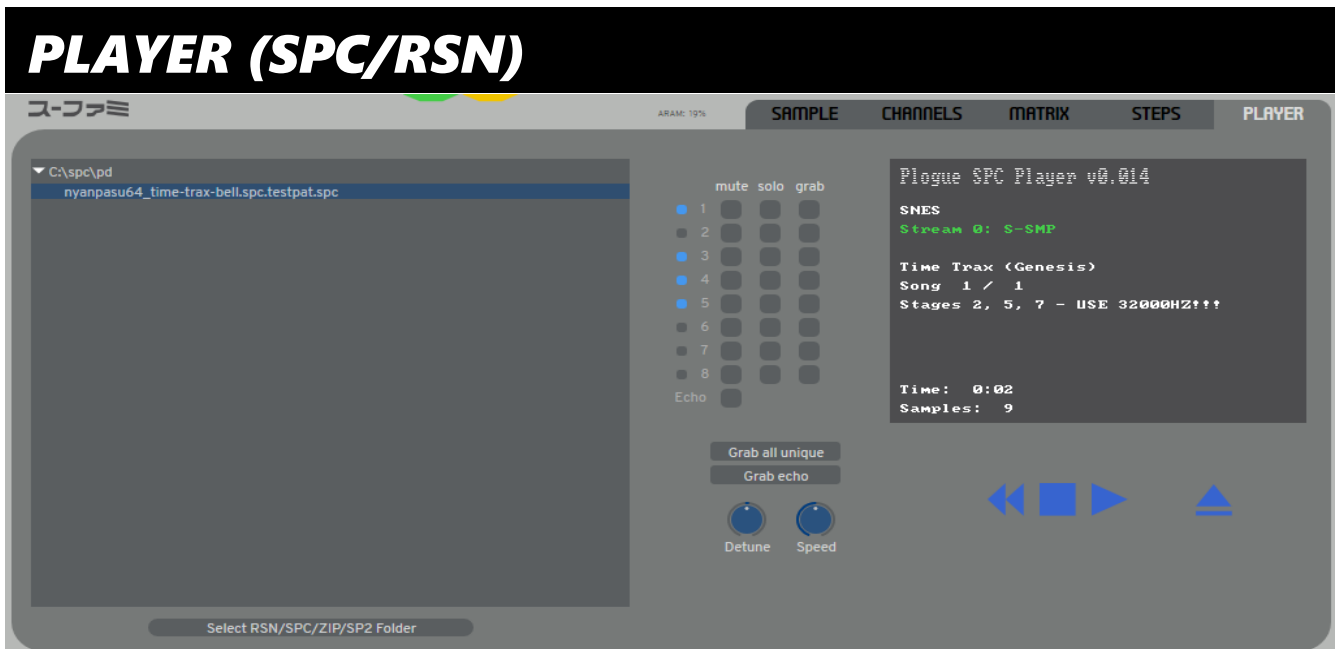
## Version and License

**Chipsynth SFC version check:** Checks if there is an updated version of chipsynth SFC.

**Fermata Engine version check:** Checks if there's an updated version of the Fermata engine.

**Show License:** Shows the key file in use (when not running in demo mode). The key file can be registered by dragging it over the chipsynth SFC interface.

## How to use chipsynth SFC



SPC files contain a RAM dump of the SPC sound module. Most games use the SPC's CPU for music, so it can be played this way. RSN/ZIP/SP2 files contain multiple SPC files compressed in an album.

**File List:** Lists the playable sound files found in the selected folder.

**Set Folder:** Selects where playable files are searched.

### Play Controls

**Previous:** Go back to previous song in multisong archive. (must be playing multi-song archive)

**Rewind:** Go back to the start of the current song.

**Stop:** Stop playback.

**Play:** Start playback.

**Next:** Advance to next song in multisong archive. (must be playing multi-song archive)

**Load:** Load a different song file or archive.

**Loop:** Activates looping playback of current song. (must be playing multi-song archive)

**Detune:** Change the audio rate from SPC playback, affecting both pitch and speed.

**Speed:** Change the SPC playback rate without changing pitch.

### Channel/Grab Controls

**Mute:** Mutes any channel or the echo effect. Great for studying songs!

**Solo:** Mutes all the other channels (the ones without the solo setting).

**Grab:** Grabs the sample data from this channel into a sample slot.

**Grab all Unique:** Grab all samples that have been played from the current song.

**Grab Echo:** Grab echo settings and set the echo effect to match.

## USING CHIPSYNTH SFC AS A STANDALONE APP

Plogue chipsynth SFC can be launched by itself and played live via MIDI keyboard or other MIDI controller. The standalone version of chipsynth SFC effectively makes your computer, audio hardware and MIDI keyboard into a virtual synthesizer that can be played independently of other programs. Unlike using it as a plug-in within a sequencer, your recording ability is limited and you can not edit your performance (though you can use various audio software programs for this).

### Launching chipsynth SFC in Standalone Mode

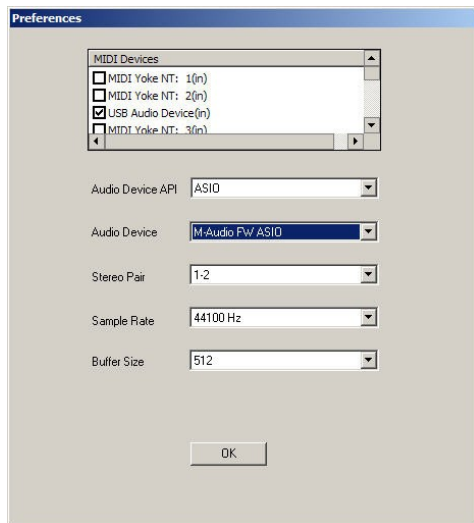
First, make sure that you have followed the instructions in the installation section of this manual. Be certain that your audio/sound interface and MIDI hardware interfaces are properly connected to the computer, your speakers or headphones are connected and everything is powered up.

To launch chipsynth SFC as a standalone application, click on the chipsynth SFC logo on your desktop or go to the Program Files or Applications folder and launch “chipsynth SFC”.

### Basic Setup Information for Standalone Mode

To use the standalone version, you have to configure the Audio and MIDI settings in the Preferences dialog box (found in the Tools menu) before you can play. When used as a plug-in, the host sequencer or tracker program has already set up its audio and MIDI connections, and the chipsynth SFC “plugs in” to them. However, with standalone operation chipsynth SFC communicates directly with your audio and MIDI interface. Setup for Mac and Windows computers is similar, except where indicated. Note that if you change your audio interface, you will almost certainly need to readjust these settings.

Open the Preferences setup dialog from the Tools menu on the chipsynth SFC standalone interface. You’ll see drop-down menus for MIDI Device, Audio Devices, Stereo Pair, Sample Rate and Buffer Size.



**MIDI Device Menu:** All supported & installed MIDI interfaces are available here. Select the desired MIDI device from the list to send and receive MIDI to it.

**Audio Device Menu:** All supported (and installed) audio interfaces are available in this drop-down list. Select the desired audio device from the list.

**Stereo Pair:** Here you can define which of the stereo outputs should be used. Many pro audio devices have a variety of outputs, so you may choose which of these you would like chipsynth SFC to output through.

**Sample Rate:** Depending on the sound card and driver you are using, various sample rates are available. Set the desired sample rate here.

**Buffer Size:** The buffer size setting will determine the delay between pressing a key on your MIDI keyboard and hearing the sound (a/k/a ‘latency’). The default buffer size of 512 samples typically

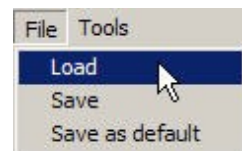
## How to use chipsynth SFC

works well, but smaller buffer sizes will give a faster response (lower latency) and higher buffer sizes will give better audio performance (more polyphony and higher fidelity). Most modern computers and audio interfaces can handle a buffer size of 512 samples without a significant reduction in polyphony. If the sound is breaking up or crackling when a note sounds, then first check that the audio connections and wiring are good. Then, try a larger audio buffer size setting. Please note that there is typically a trade-off between higher buffer sizes (polyphony and sound fidelity) and lower buffer sizes (faster response or lower latency). Also note that the sound card buffer size settings determine latency, rather than chipsynth SFC Player itself.

Once you have your Audio and MIDI set up, and have loaded one of the snapshots, you can begin playing chipsynth SFC. Try playing a key on your MIDI keyboard. If the MIDI and audio configurations are correct, you should hear the corresponding synth note. If not, check the MIDI connections and wiring, and the MIDI output channel of your MIDI keyboard. Also check that the channel is specified correctly. If you are hearing the notes play, then the basic configuration is complete, and you are ready to use chipsynth SFC.

### File Menu for Loading and Saving Snapshots in the Standalone

Configuration presets (.fermatax files) for chipsynth SFC can be saved and loaded. This gives the user the ability to customize instruments setups to suit personal preferences and save configurations for convenient future use. The File menu choices are:

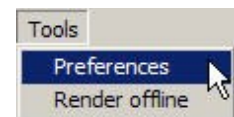


- ◆ **Load**—any saved configuration preset files in Fermata format can be loaded by clicking on this choice and selecting the desired file.
- ◆ **Save**—any configuration can be saved by clicking on this choice, typing a name for the custom preset and saving to a desired location.
- ◆ **Save as default**—any settings can be saved as part of the default, to be loaded automatically at the time the chipsynth SFC is booted in standalone mode.

### Tools Menu in Standalone ONLY

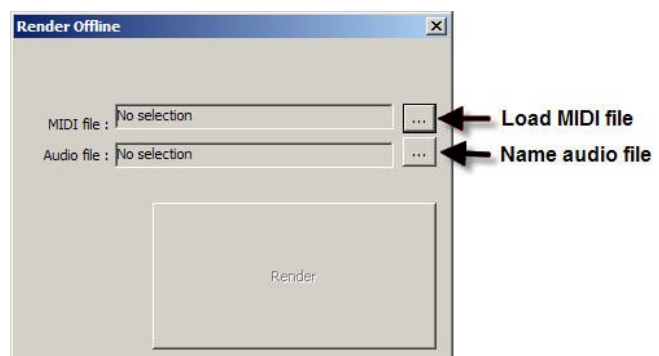
The Tools menu includes:

- ◆ **Preferences**—as described in the basic setup information above.
- ◆ **Render offline**—MIDI files can be rendered to audio offline using this feature.



To use the Render Offline feature:

1. Click on the Load MIDI file button.
2. Select the desired MIDI file
3. Click on the Name audio file button.
4. Name the audio file and specify its location.
5. Once the files are in place, click on the Render button, which will render the files to the selected audio file.






## USING CHIPSYNTH SFC AS A PLUG-IN

When used as a plug-in, chipsynth SFC is not a standalone program, but rather a virtual instrument module that seamlessly integrates into your favorite music software program, sequencer or tracker (the “host”, which is modular in order to accept plug-ins).

Using chipsynth SFC as a plug-in has various advantages:

- ◆ MIDI recording and sequencing
- ◆ Easy automation of parameters through the use of MIDI CCs or the host's automation.
- ◆ Audio mixing with other plug-ins and effect processing of chipsynth SFC within a single program.
- ◆ Integration with other instruments into a “virtual studio”.

A great thing about plug-ins is that they work with a large variety of compatible music programs: chipsynth SFC can be used as a VST plug-in in many VST music programs, sequencers, supported tracker programs and hosts. It can also be used as an Audio Unit or AAX plug-in.

Plug-in Standard	Description	Windows 64-bit	Mac 64-bit
VST2, VST3 	VST (Virtual Studio Technology) was developed by Steinberg for Cubase, and is also used by Ableton Live, FL Studio, Studio One, Reason, Bidule, etc.	X	X
Audio Units 	The Audio Units (AU) plug-in standard was developed by Apple for Core Audio in Mac OS X. Audio Units is the preferred plug-in format on Mac, used by Apple GarageBand & Logic and MOTU Digital Performer.		X
AAX 	AAX plug-ins are designed for Digidesign Pro Tools, used extensively in the pro audio and post production communities.	X	X

### Plug-in Use

To use chipsynth SFC as a plug-in instrument, you simply launch your host music application/sequencer first and then launch chipsynth SFC from within it. Make sure that your sequencing host program is properly installed and configured, and that it is producing sound properly. Used as a plug-in, chipsynth SFC's audio and MIDI data is managed by the host music software.

Each music software application has its own approach to handling plug-in instruments. They each have a different method of installation as well as differing means of loading and accessing plug-ins. It is important to make sure that you refer to the instructions in your music software application's manual regarding the loading and operation of plug-in instruments.

### Windows VST Setup

Make sure the chipsynth SFC plugin is within the VST plugin folder used by your host, so that it can be scanned and recognized on startup.



## CREDITS

### Plogue chipsynth SFC

Copyright © 2020 Plogue Art et Technologie, Inc. All rights reserved.

Produced by	David Viens
FERMATA Engine Design/Development	Plogue Art et Technologie, Inc.
Hardware research and modeling	David Viens Hubert Lamontagne
Main testing and suggestions	Mathew Valente [TSSF]
Presets	David Viens, Hubert Lamontagne, Mathew Valente, Xenos Soundworks, John Montoya
UX early concepts and logo	JCPP
Additional UX help	A_Rival, Chris Randall

#### Acknowledgments

Mathew Valente [TSSF]: This software would not be what it is without your tireless dedication and your love of the system.

And in no particular order:

Artemio Urbina	Wohali	Infinite NES Lives
Jonathan Gevanyahu (LN)	Kevin Horton (kevtris)	Cuckoo
Matt Gallais (pixoshiru)	Shay Green (Blargg)	madscient
Akira Key	D. Smolken	Alex Hauptmann
Johnny Collazo Jr.	Peter Swimm	Inverse Phase
Jérôme LeBel	VECTRAX	Andy Lemon
XC3N	Xenos Soundworks	David Rysk.
John Montoya	Aaron Giles	Shannon Mason
Fanny Rebillard	Nicola Salmoria	R. Belmont
Tim Swartz	David Wise	Chris Randall
Steffan Andrews	Toy Company Crew	Dominic Arseneault

Thank you!

The information in this document is subject to change without notice and does not represent a commitment on the part of Plogue Art et Technologie, Inc. No part of this publication may be copied, reproduced or otherwise transmitted or recorded, for any purpose, without prior written permission by Plogue Art et Technologie, Inc.

Plogue chipsynth SFC® is a registered trademark of Plogue Art et Technologie, Inc. Use of Plogue chipsynth SFC® software instrument and the contents herein are subject to the terms and conditions of the license agreement distributed with the program (see license.rtf). You should carefully read the license agreement before using this product. The programming and presets presented in Plogue chipsynth SFC® are protected by copyright and cannot be redistributed, whether modified or unmodified. The Guide to Plogue chipsynth SFC contained herein is also covered by copyright.

Any other trademarks of third-party programs are trademarks of their respective owners. The information contained herein may change without notice and does not represent a commitment on the part of Plogue Art et Technologie, Inc.